California Franchise Tax Board (FTB)

Enterprise Architecture Definition
Service Oriented Architecture (SOA)

Version No. 1.1

April 4, 2008
Author: Enterprise Architecture

Document Information

Document Source

This document is controlled through Document and Deliverable Management. To verify that this document is the latest version, contact Enterprise Architecture.

Revision History

Version No.	Date	Summary of Changes	Revision Marks
1.1	2/05/2008	Formatting changes, reference change (NIST 800-95)	John R.

Table of Contents

List of Figures

1.0 Executive Summary and Charter

1.1 Overview

The Franchise Tax Board (FTB) and the State of California have a long-term vision of moving to a Service Oriented Architecture (SOA) enterprise environment, which will provide better utilization of functionality and reduce the development of redundant systems across the enterprise. In an SOA, business process functionality and data are exposed as a service callable by multiple applications, thus maximizing the cost-effectiveness of developing and implementing the service. In order to be useful, the service will perform a small unit of functionality that is flexible so callers will be able to integrate the service with other services to perform the required functions of an application. To be used effectively by the enterprise, these services will be described, discoverable and useable without regards to the system, program or organizational unit.

There are three basic types of enterprise-wide services under the SOA umbrella: the Common Business Services, the Common Infrastructure Services, and the Common Information or Data Services.

A Common Business Service is a common business function exposed as a service (web service) that provides business value to more than one system using Service Oriented Architecture (SOA). The Common Business Services provide business functionality to enterprise consumers such as Address, Locate, Noticing, etc.

The Common Infrastructure Services provide basic infrastructure services to consumers such as single audit logging, error handling, and security, etc.

The Common Information or Data Services provide data to consumers from data repositories.

The combination of web services and services, internal and external to an organization make up a service-oriented architecture.

1.2 Scope

The SOA architecture definition defines the current and target states of FTB's Service Oriented architecture, a gap analysis and a strategy for implementation. The following list contains the subject areas covered:

- Governance to support SOA
- Common business services, common infrastructure services and common data services
- SOA enabling technologies

The common business services, common infrastructure services and common data services are dependent on other core areas such as Security, Data Management and Delivery and Content Management, which have their own Architectural Definition. There are many dependencies across the enterprise that must be aligned in order for any one architectural area to provide a cost effective and efficient solution. This Architecture Definition Document will focus on SOA from a business and technology perspective.

1.3 High-level Requirements

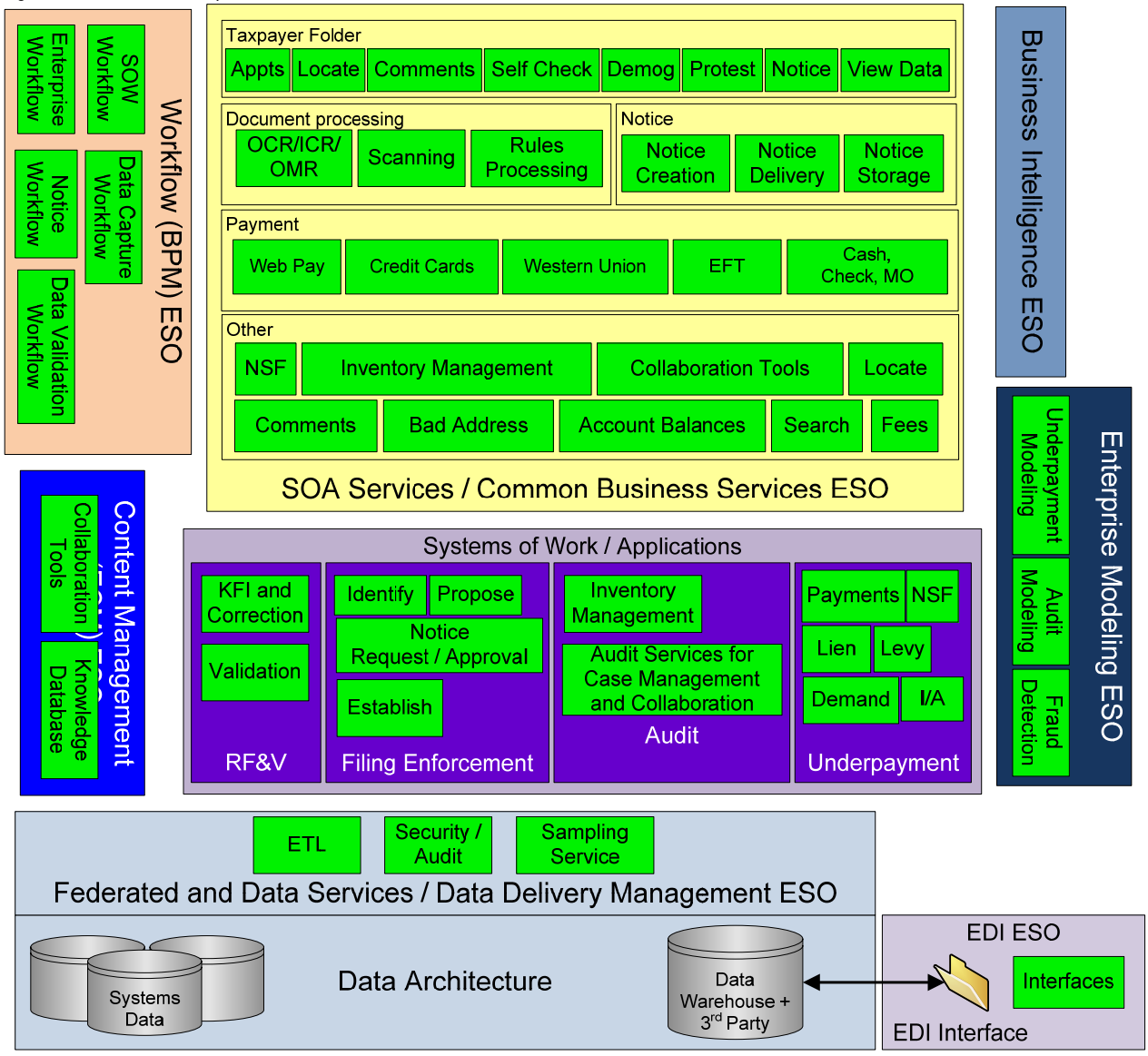
The following table outlines the high-level requirements of SOA.

Figure 1.3-1: SOA – High Level Requirements

Requirement	
Reusable	Services should be reusable by multiple clients to maximize the investment and utilization of the service.
Scalable	Services should scale well for the particular applications. Use of products such as Microsoft Word to render documents should be avoided.
Securable	Services should implement standards-based security models.
Standards Based	Services should use standards-based communication protocols.
Available	Services should be implemented in an environment that can provide continued service in the event of component failures. Services should meet requirements of Service Level Agreements.
Maintainable	Services should be built in ways that minimize the time necessary to implement changes and legislative mandates. A method of providing service versions to callers is desirable.
Discoverable	A means for enterprise users to identify and understanding the available services should be implemented.

1.4 Conceptual Architecture

Figure 1.4-1: Future Conceptual Architecture



2.0 Current Architecture

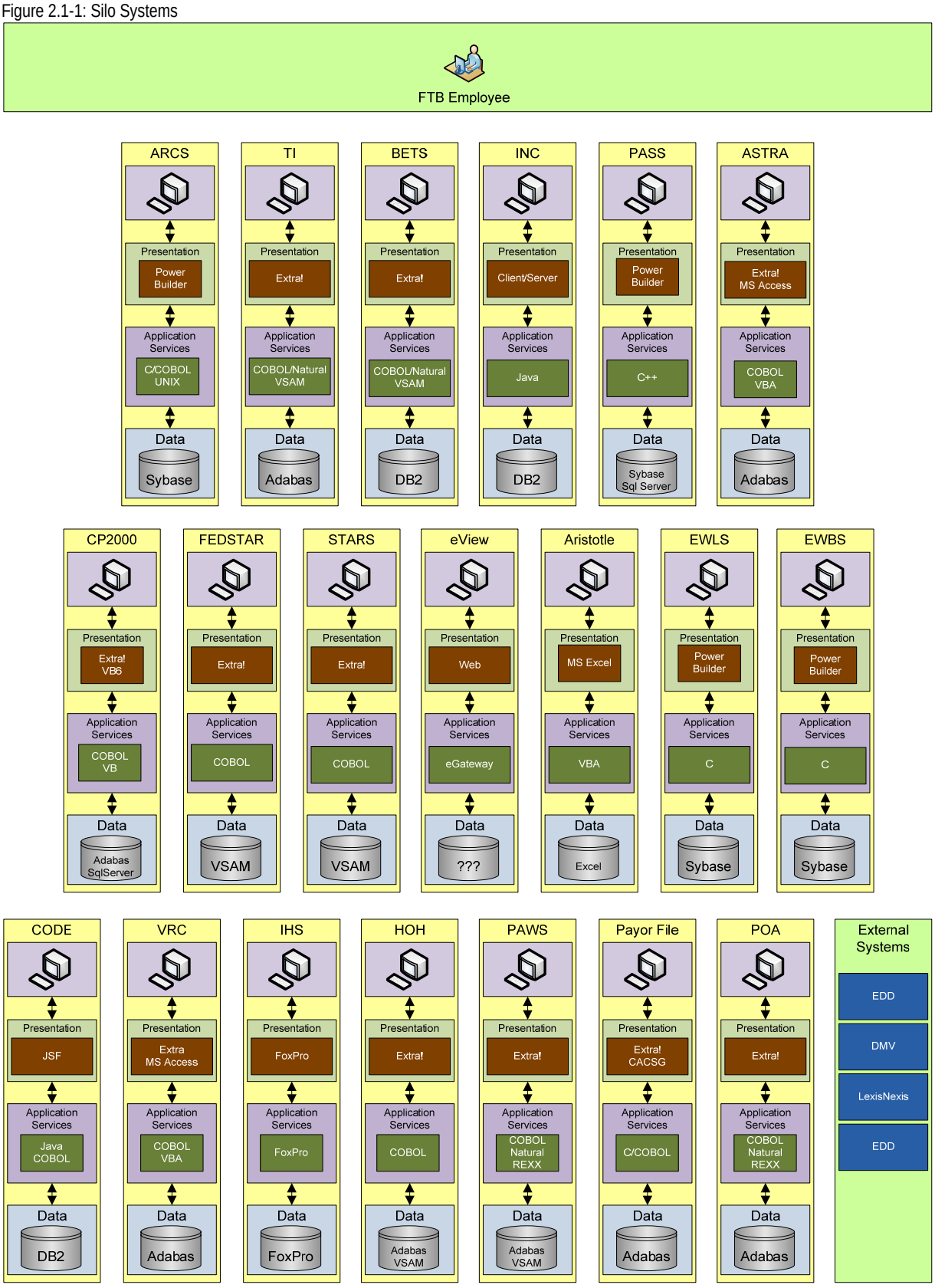
FTB's current architecture has two types of systems; "silo" stand alone and Web Service based.

2.1 Silo Systems

Application development at FTB has followed a 'silo' architecture where applications are not designed to work together to support common enterprise business processes but to support a functional area's needs. Advances in hardware and software technology coupled with the business need for more information and quicker turnaround times have made it necessary for FTB to change its approach to building systems.

In the figure below, FTB's major systems are shown. Utilizing a traditional, application-centric approach to application development, each system has a dedicated database, hardware environment, and tightly coupled application modules. Each area's applications were written in different languages, many times based on vendor preference. Data residing in one system needed by another system has often been replicated. Today FTB supports over 53 programming languages, 55 databases and 7 operating systems.

Figure 2.1-1: Silo Systems



2.2 Enterprise Web Services

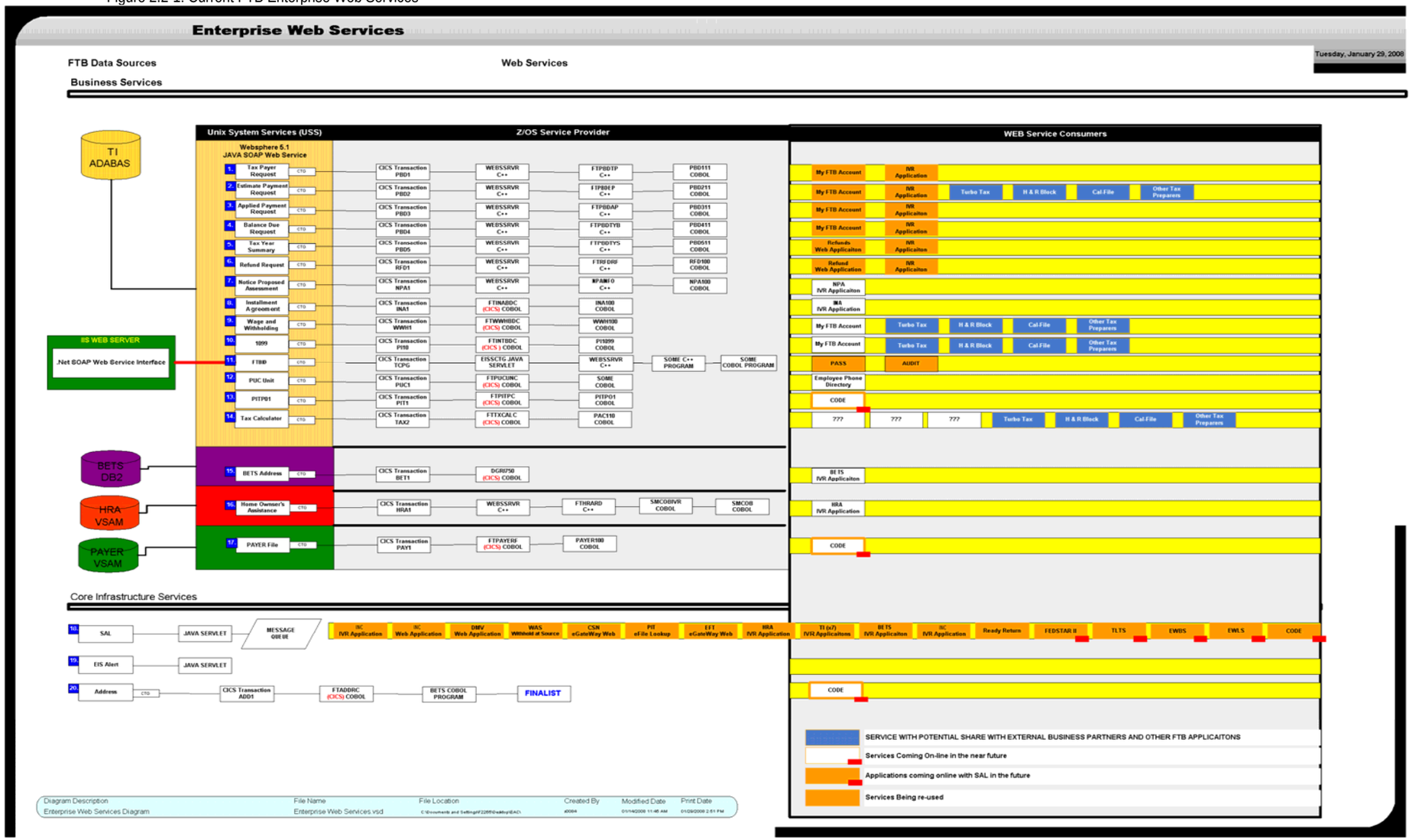
In the early 1990's, with FTB's introduction of its Interactive Voice Response (IVR) system and public facing web applications, it began to leverage back end legacy data and business rules from the Taxpayer Information system (TI) via screen scraping. Screen scraping simulates the steps that an actual user of the system would take to obtain the required data. In the late 1990s, the first web services were written to extract data directly from TI for the Payment and Balance Due and Refund applications. This eliminated production problems that resulted from changes made to the TI screens that were being screen scraped. A new infrastructure was created that allowed FTB to leverage back end application logic and provide this information to the IVR and the Internet through the use of web services and XML.

Today, FTB has a library of web services that foster reuse of existing business logic. The current systems being accessed with web services technology have expanded beyond TI and now include the Business Entity Tax System (BETS), eGateway, Integrated None-filer Compliance system (INC) and Homeowner and Renters Assistance System (HRA).

The figure below shows FTB's library of web services, which foster reuse of existing business logic. On the right side of the figure, the Web Service consumers are listed. The consumers call the appropriate business service that resides on WebSphere. Back end application logic is leveraged from the Z/OS service providers for business rules and data access. Utilizing XML, the results are returned to the Web Service consumer.

Core infrastructure services are listed at the bottom of the diagram. The security audit logging service (SAL), is called by the business services to perform audit logging.

Figure 2.2-1: Current FTB Enterprise Web Services

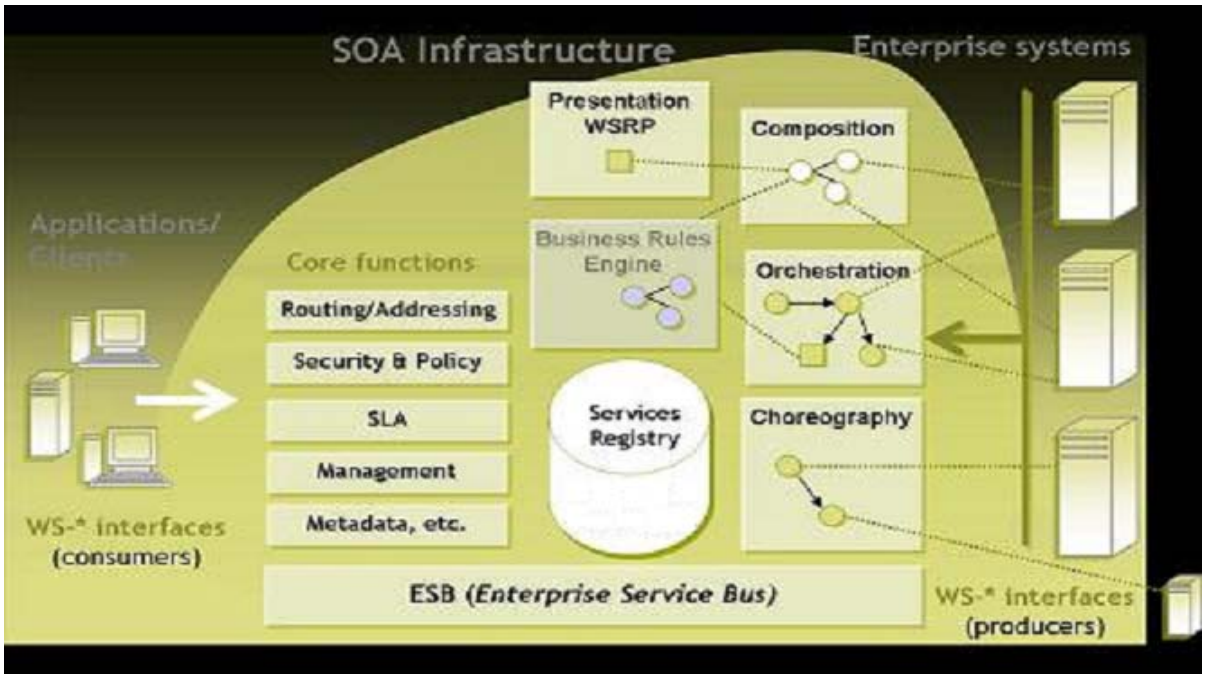


3.0 Target Architecture

3.1 Future Capabilities and Components

SOA is a business-centric IT architectural approach to integrate business processes through the implementation of repeatable tasks. FTB will expand its use of services to meet the business needs of the enterprise and promote reusability. The figure below represents a mature SOA infrastructure.

Figure 3.1-1: High Level Mature SOA Infrastructure



3.1.1 Web Services

Web Services at FTB can include business logic, business rules, and data. These Web Services will be available to systems within and outside of FTB. Web Services will be written in either Java or .NET, incorporate a web service interface, will use SOAP messaging, will be designed to be interoperable and not machine dependent and will not use operating platform specific API's. Any operating system inside FTB that can communicate using the standard HTTP protocol or Message Queuing technologies will be able to send and receive information using web services. FTBs Web Services may be hosted and executed on any operating system. These Web Services will communicate with a client using industry standard XML messages that follow the SOAP-standard. Common in both the Web Services field and industry it is standard that each web service has a Web Service Definition Language (WSDL) file. The WSDL file itself is a prerequisite for automated client-side code generation in the mainstream Java and .NET SOAP frameworks. FTB will mandate both SOAP and WSDL in their definition of a web service ensuring interoperability within the FTB organization, the California State Enterprise Architecture Program (CEAP), and private industry customers. Exposing web services to external customers may also generate new revenue streams for FTB in the future. Below are service types FTB will use to define its architecture.

Figure 3.1-2: FTB service types

Service	Description
Business Service	FTB currently and will continue to use Business Services. A business service is the logical encapsulation of a business function. The following are examples of a business service. A tax calculation of penalties and interest. An estimated payment calculation for a taxpayer.
Data Service	Will be capable of delivering data from any type of information store. It will not matter what company or format the database is in or from.
Core Service or Infrastructure Service	Are the “plumbing” services. These services are leveraged to increase the sophistication with which ESB is able to carry out messaging, routing, and SOA related functions. Security and business policies will need to incorporate or introduce rules. Some of the cores services may be applied together with policy services and security centralization.
Centralized Rule Service	Most often classified as members of the core services layer because they provide generic processing functionality leveraging technology resources. Their functional context is not derived from any organization-specific business models. Even though rule data is business-centric, to the rules service, it is just data that it is required to manage and dispense.
Service Management Service	Assist with managing an SOA environment by providing mechanisms to install, maintain, monitor, and troubleshoot Web Services.
Service Communication Service	Provides support for various types of communications models between services: queued messaging, publish-subscribe event notification, and distributed logging services.
Policy Service	Provides a framework for creating, administering and managing policies for the SOA infrastructure; these policies cover security, resource allocation, and performance.
Security Service	Provides support for different security models, mechanisms, protocols and technologies that extend core Web Service security protocols. They support activities such as authorization, authentication, trust policy enforcement and credential transformation.
Business Rules Service	Physically abstracted into a dedicated part of the architecture under the management of specialized rules engines and platforms. This centralizes access to business rule logic and avoids redundancy. It further centralizes the governance of business rules so that they can be modified and evolved from a single location. See BPM Architecture Definition Document

3.1.2 Services Registry and Repository

FTB will implement a registry with full support for the Universal Description, Discovery and Integration (UDDI) protocols. The registry will provide a single point of reference for developers to register, discover, and govern web services. This Web Service registry, like other web service components, will be standards-based to foster interoperability across organizational boundaries. The registry will maintain basic information about a service and will provide links to service metadata and artifacts (that will be stored in the repository).

3.1.2.1 Standards Enforcement

The registration process will provide a point of control at which FTB can perform governance compliance tests and institute basic approval processes during service configuration and release management. The registry will play an important role as part of the runtime governance infrastructure providing a single point of reference for all service information enabling service endpoints and intermediaries to share information.

The FTB will define common standards to be followed by every service provider by the SOA center of Excellence (COE) (please see SOA governance section). The service registry administrator (SRA) will ensure that FTB enterprise architecture SOA standards have been applied. These standards will follow industry best practices.

3.1.2.2 Publishing and identifying services

Having each service published in a common services repository will allow for the discovery of the existence and location of services. This ensures FTB employees will be able to determine which services exist, and what the services do within the enterprise organization. The repository will contain functionality meta-data that will describe the data elements that the service can return and describe how the service can be called or invoked.

3.1.2.3 Monitoring, logging and tracking services

Because of security concerns, governmental regulations dictate monitoring, logging and tracking standards. FTB will establish an effective monitoring, logging, and service tracking system to track which services are being used, how often and by whom. Tracking of information is crucial for future reference and audit events.

3.1.2.4 Service Level Agreements

FTB will specify and enforce Service Level Agreements (SLA) between consumers and producers of Web Services in the registry. By centrally publishing service information to the registry, all potential users of the service will be able to discover it easily. During the establishment of the SLA, service consumers and producers (service providers) negotiate a utilization contract. This process involves the following steps:

- The service consumer applies for permission to use a service.
- The service consumer and service provider negotiate acceptable levels of service and other issues.
- Any agreements that impact the runtime infrastructure are propagated to the appropriate service mediation system for runtime enforcement.
- The service consumer is provisioned to use the service.

3.1.2.5 Transparency

A service registry utilizing an Enterprise Service Bus (ESB) will provide Web Services transparency. All clients of a Web Service will “point” to the ESB. The service registry “knows” the details (such as location and interface) of the web service and the ESB “consults with” the service registry to determine where to route the service request. This provides flexibility as the web services can be updated and moved without affecting the users.

3.1.3 Enterprise Service Bus (ESB)

The ESB will become the backbone used to deliver web services at FTB. In order for FTB’s SOA infrastructure to be dependable, robust and secure, it will be necessary to connect any IT resource, regardless of technology or location. The ESB will easily combine and re-assemble service dependencies. FTB will determine, within the SOA CoE, if SOA deployments will solely rely on the use of an ESB, or just utilize an ESB to offload non-core processing tasks.

The ESB and service registry must be scalable with high availability, and have a well-designed recovery plan in case of a disaster. The FTB’s ESB will have the following features:

3.1.3.1 Dynamic Lookup and Routing

FTB’s ESB will support "virtual services." The ESB appears to be the real service provider to the requester, routing messages to the actual service provider. FTB will store the location of a service provider in an external registry, using WSDL files. The ESB will look up services at run time. The endpoint information will be managed centrally, as part of an overall SOA governance model.

3.1.3.2 Content-based Routing

Content-based routing is a special case dynamic routing. Lookups will occur based on criteria that can be found inside the web service SOAP message and is based on content and context of XML messages.

3.1.3.3 Message Aggregation and Distribution

In most scenarios in FTB’s SOA today, one service requester invokes one web service. FTB’s future SOA environment will have a one-to-many relationship. For example, a requester sends a request, resulting in multiple web services be orchestrated together and a single aggregated response is sent back. Another scenario is a request sends a request, and that request is sent to multiple service providers. Once one of the multiple service providers have responded, the web service response being sent back to the requestor is aggregated into one consolidated response message.

3.1.3.4 Message Transformation

Most messages in the ESB will be XML-based and not all messages in the ESB will require transformation. The ESB will have the ability to use XSLT transformation on messages flowing through it. Plug-ins will be available to provide support for very complex transformations and offers an API that can be invoked during the ESB transformation.

3.1.3.5 Messaging Infrastructure

An ESB may be tied to whatever messaging infrastructure (MOM), that FTB supports (WebSphere MQ), or the ESB may allow for adapters using JCA technology. However, most

ESB vendor products are building upon MOM messaging infrastructures. Which messaging service we use is dependent on the tool that is selected.

3.1.3.6 Protocol Support

FTB's ESB will be multi-protocol where varieties of protocols (WS-SOAP, JMS, JCA, etc.) natively interact with the ESB, without employing an adapter. This is "how you get on the bus." Requestors using one protocol can invoke services that are exposed using a different protocol. It is also possible to support different security protocols. For example, FTB could perform a SAML based authentication and authorization with a web service consumer from the Department of Technology Services and then the ESB would convert that authentication header to a format accepted by a backend web service requiring basic authentication over SSL on the mainframe.

3.1.3.7 Adapters

The ESB will have adapters to provide connectivity to all internal and external. This is needed for legacy systems not built with a messaging model.

The ESB will transform messages into a legacy format that is understandable by the application. The software responsible for effecting these transformations is referred to as an adapter.

3.1.3.8 Security

With the implementation of the ESB, FTB will incrementally implement the new authentication and authorization web services. The ESB can map security to existing security mechanisms that are already in place, and over time utilize the new security mechanisms as necessary without disruption to the end customers of the services. For example, FTB implemented a web service to access TI through WebSphere. This Web Service required basic authentication credentials. It was later decided FTB wanted to eliminate WebSphere, and add a web service implemented instead in CICS 3.1. With this scenario, the clients would simply call the web service on the ESB. However, the ESB would be modified to change the Web Service call to CICS directly and to send the appropriate security credentials as needed. The ESB would map the web service authentication and authorization scheme to CICS authentication from the container manage security enforce by the WebSphere implementation.

3.1.4 Business Rule Engines

FTB will have a Business Rule Engine (BRE), which is a software system that helps manage and automate FTB business rules. The rules that will be defined in a BRE may come from business areas, legal regulations, FTB policy, security, or other sources. BRE engines are pluggable software components that separate the business rules from the web services and application code. This allows the business users to modify the rules frequently without the need of IT and allows the services applications to be more adaptable to business agility. A BRE Engines should allow FTB to implement dynamic rules in business processes without having re-program business services. (Please see the BMP Enterprise Architecture Definition Document.)

3.2 Future Enterprise Governance

SOA governance will organize and align efforts to manage and control the enterprise architecture realizing the objective of Service Oriented Architecture. FTB SOA goals are to:

- Align business and IT

- Establish service re-use
- Establish an agile service environment

FTB's SOA governance will define the interrelationships of different groups, participants and services and how they will work as a cohesive unit within the larger SOA architecture.

In order for the FTB to develop a mature SOA enterprise architecture, it is necessary to align business and IT to a methodology and process that connects business and enterprise architecture. To achieve services reuse, FTB will ensure the existence of standards put into place an effective governance model. FTB will achieve agility through the ability to program IT processes at a higher level of abstraction than raw services. Service orchestration or BPM workflow with configurable policies and SLA/OLA contracts will be in place.

3.2.1 Center of Excellence (CoE)

FTB is following the emerging trend of establishing a SOA CoE. The CoE will be an organizational unit that facilitates an exchange of ideas between business and IT leaders and experts. The CoE gathers input from multi-disciplinary skill sets, which operate across organizational boundaries in order to make shared technology decisions. The SOA architecture is developed, documented, and communicated to the enterprise from the CoE. The CoE enforces compliance of standards, ensures highly tuned processes, and drives software and data re-use. The result is less development cost, less testing and lower support costs.

3.2.1.1 Reporting

The CoE will be involved in compliance reporting to management and information security.

3.2.1.2 Requirements and Policies

The CoE will develop, document and communicate:

- Service Interface Specifications
- Service Security Requirements
- Information Model
- Service Architecture Specifications
- Service Architecture Diagrams

3.2.1.3 Guides and Checklists

The CoE will develop and disseminate service question and answer checklists, service quick start guides, and deployment checklists for SOA system software products.

3.2.1.4 SOA Training

The CoE will help FTB define what technology training that FTB will invest. The CoE will have executive support to achieve cross organization cooperation.

3.2.2 Service Registry Administrator (SRA)

The Service Registry Administrator (SRA) manages the consistency of the catalog and enforces guidelines that protect against redundancy, proliferation and unauthorized modifications of the service catalog. The SRA controls the adoption of service definitions into the shared registry,

and is responsible for controlling consistency and quality of the design of services, including the service-data relationship. The SRA is a member of the CoE.

4.0 Gap Analysis

FTB, like many organizations, has developed silo-based systems where, focus is solely on the business process it performs. This has created an IT infrastructure with the following problems:
Dedicated silo hardware that is expensive: FTB's applications run on dedicated application server and database computers. This equipment is typically provisioned to handle the worst-case load, and is usually highly underutilized.

Synchronizing silo data is complex and error-prone: Most FTB applications have their own operational data stores, thereby creating a complex data synchronization infrastructure, particularly for shared data about products, partners, and customers. It's almost impossible to get a centralized view of the data here at FTB.

Integrating silo applications is difficult: Getting silo applications to integrate is an ongoing challenge, particularly when the underlying reference data between two silos is not in sync.

Staff assigned to and focused on a specific process within one System of Work (SOW): This type of structure is not conducive to collaboration between systems, and sharing of data and knowledge.

To solve these gaps, FTB's IT will implement cultural changes, create service registry and repository, establish service certification and ownership, implement version control, create service level agreements, create SOA architecture and implement an SOA security infrastructure (See IAM Architecture Definition Document), and re-align (See IT Strategic Plan)

4.1 Cultural Changes

Current IT efforts are focused on delivering applications as quickly as possible at the lowest possible cost. Organizational structure, accounting practices, and incentive systems all reinforce this goal. FTB must develop a culture where external solutions are understood and used. This will include:

- Adopting different ways of working and different ways of thinking, which include cross organization cooperation and creating news roles within the organization with different responsibilities. The newly created Operations bureau and the SOA CoE are the start of this process.
- Foster a technical and cultural environment where reuse is considered a characteristic of excellence in software engineering.
- Facilitate reuse of services through communication, leadership and governance.
- Focus on long-term goals rather than individual project costs and timelines. Implement the best, most cost effective long-term solutions.
- Prevent silos by preventing application bureaus from operating independently.
- Establish a management group to prioritize cross-functional application enhancements and annual changes that follow EA recommendations and provide the best value for FTB.
- Provide answers for the following questions:

- Who pays for the service infrastructure?
- Who pays for initial service development?
- How are costs shared across service consumers?
- Who pays for service enhancements?
- Who pays for upgrades to the infrastructure to support rising load on a service and?

4.2 Service Registry and Repository

Today FTB does not have a registry and repository for web services causing duplication of effort. FTB must establish a repository to make visible enterprise services. Reuse of services will be facilitated through governance and a well-described service repository that describes, classifies, and makes discoverability possible.

4.3 Service Certification and Ownership

Today, FTB does not have governance for enterprise service certification. Governance will need to be created to manage shared services, make service ownership determinations and policies for modifying, extending, combining, or retiring a service. The certification process must be defined and published so developers understand and use the process to get their new services certified.

4.4 Version Control

Today there is no version control for enterprise web services. As the number of services expands, lack of version control will lead to "legacy SOA applications." Deploying a new version of the interface may require changes to all clients of the previous version. Without well-established SOA management, not all users of the services are known. Software versioning is a requirement of the SOA FTB environment.

4.5 Service Level Agreements (SLA)

FTB does not have contracts between service consumers and service providers. As more consumers use FTB services, it will be necessary to define who the consumers of a service are and the acceptable levels of service for each consumer. For example, when sharing a service with an external agency or outside vendor, their acceptable level of service may be different than what an FTB user of the service may negotiate. Internally at FTB different areas have different levels of acceptable service. Through the SLA, FTB can manage our application and network resources and how they are being used. Automated processes will be set up to notify FTB when acceptable levels of service are in danger or being compromised and to enforce the SLA by redirection of resources. Consumers of a service will have their own SLA with the service provider. For example, a service that calculates tax with penalties and interest might be invoked by many different applications. Due to the financial nature of the tax calculation routine, it would be reasonable to expect a minimum level of service. If the maximum expected response time for the tax calculation routine is 400 ms, any scenario that a response time exceeding 400ms might be indicative of a problem and be dealt with accordingly. Unless a prior contract exists between the consumer and the tax calculation service, there would be no way to measure

and enforce such a service level agreement. The directory will be a place to communicate and share SLA agreements and use technology to enforce them.

4.6 SOA Infrastructure

FTB has a limited SOA infrastructure. Tools and processes will be implemented to advance from our current state.

4.7 Security

FTB has no security policy that will support an advanced SOA infrastructure. Exposing web services and data to the Internet, is a security concern. The industry has many implementation standards for web services security that can be implemented to address the security aspects of sharing data with external customers and business partners (see the IAM Architecture Definition Document).

5.0 Roadmap

The following chart illustrates how SOA will be implemented at FTB.

Figure 4.7-1: Service Oriented Architecture Phases

Task Name

Service Oriented Architecture: Common Business Services

PART 1 - Service Oriented Architecture: Planning and Governance

SOA Planning

Coordinate with other ESOs in Data Governance and Standards Planning

Service Monitoring Governance and Standards
- Determine Requirements for Availability, Logging, Auditing, Performance Metrics
- Document Service Monitoring Governance and Standards
- Approve Service Monitoring Governance and Standards
- Implement Service Monitoring Governance and Standards
- Communication

Exception Management Governance and Standards
- Determine Requirements for Exception Management and Error Handling
- Document Exception Management Governance and Standards
- Approve Exception Handling Governance and Standards
- Implement Error Handling Governance and Standards
- Communication

Service Delivery Governance and Standards
- Determine Requirements for Version Management
- Document Version Management Governance and Standards
- Approve Version Management Governance and Standards
- Implement Version Management Governance and Standards
- Communication

Version Management Governance and Standards
- Determine Requirements for Version Management
- Document Version Management Governance and Standards
- Approve Version Management Governance and Standards
- Implement Version Management Governance and Standards
- Communication

Service Level Agreements (SLAs) Governance and Standards
- Determine Requirements for SLAs
- Document SLA Governance and Standards
- Approve SLA Governance and Standards
- Implement SLA Governance and Standards
- Communication

Service Registry Governance and Standards
- Determine Requirements for the Services Registry & Repository
- Document Registry Governance and Standards
- Approve Registry Governance and Standards
- Implement Registry Governance and Standards
- Communication

PART 2 - Establish SOA Service Registry

Select Service Repository
Implement Service Registry Tool
Test Registry
Gather Service Information & Populate Registry
Milestone - SOA Registry Established

PART 3 - Security and ID Management Services

Collaborate with Security to establish IAM infrastructure requirements for services:
- Identity Services
- Authentication Services
- Authorization and Privacy Services
- Confidentiality and Integrity Services

Hours are not included for development & testing of these services - assuming they are being purchased and included in the Security estimates.

PART 4 - Enterprise Service Bus

Determine Requirements for the ESB
Select ESB
Implement ESB
Test ESB
Communication
Milestone -ESB Established

PART 5 - SOA: Establish Data Subject Area Services - Collaboration with DDM

Establish Data Subject Area Services for CUSTOMER/PARTY

Review Data Access Requirements for Data/Web Services
Design Data/Web Services
Code Data/Web Services
Unit Test Data/Web Services
Implement/Document Data/Web Services
Support System/Integration Test

Establish Data Subject Area Services for CUSTOMER ACCOUNT

Review Data Access Requirements for Data/Web Services
Design Data/Web Services
Code Data/Web Services Requirements have been met
Unit Test Data/Web Services
Implement/Document Data/Web Services
Support System/Integration Test

Establish Data Subject Area Services for TAX DECLARATION

Review Data Access Requirements for Data/Web Services
Design Data/Web Services
Code Data/Web Services
Unit Test Data/Web Services
Implement/Document Data/Web Services
Support System/Integration Test

Establish Data Subject Area Services for ASSET and INCOME

Review Data Access Requirements for Data/Web Services
Design Data/Web Services
Code Data/Web Services
Unit Test Data/Web Services
Implement/Document Data/Web Services
Support System/Integration Test

Establish Data Subject Area Services for CUSTOMER COMMUNICATION

Review Data Access Requirements for Data/Web Services
Design Data/Web Services
Code Data/Web Services
Unit Test Data/Web Services
Implement/Document Data/Web Services
Support System/Integration Test

PART 6A - Establish Underpayment Modeling Service

Review Business Requirements
Design
Code
Unit Test
Implement/Document
Support System/Integration System

PART 6B - Establish Underpayment Data/Web Services - Collaboration BI & DDM

Review Data Access Requirements for Data/Web Services
Design Data/Web Services
Code Data/Web Services
Unit Test Data/Web Services
Implement/Document Data/Web Services
Support System/Integration Testing

PART 7 - Establish Filing Enforcement (FE) Data/Web Services - Collaboration BI & DDM

Review Data Access Requirements for Data/Web Services
Design Data/Web Services
Code Data/Web Services
Unit Test Data/Web Services
Implement/Document Data/Web Services
Support System/Integration Testing

PART 8 - Establish Audit Data/Web Services - Collaboration with BI & DDM

Review Data Access Requirements for Data/Web Services
Design Data/Web Services
Code Data/Web Services
Unit Test Data/Web Services
Implement/Document Data/Web Services
Support System/Integration Testing

PART 9 - Establish Fraud Data/Web Services - Collaboration with BI & DDM

Review Data Access Requirements for Data/Web Services
Design Data/Web Services
Code Data/Web Services
Unit Test Data/Web Services for Performance
Implement/Document Data/Web Services
Support System/Integration Testing

PART 10A - Establish ADDRESS Service

Review Business Requirements
Design
Code
Unit Test
Implement/Document
Support System/Integration Testing

PART 10B - Establish ADDRESS Data/Web Services - Collaboration with DDM

Review Data Access Requirements for Data/Web Services
Design Data/Web Services
Code Data/Web Services
Unit Test Data/Web Services
Implement/Document Data/Web Services
Support System/Integration Testing

PART 11A - Establish Enterprise Noticing Service

Review Business Requirements
Analysis and Design
Code
Unit Test
Implement/Document
Support System/Integration Testing

PART 11B - Establish ENTERPRISE NOTICING Data/Web Services - Collaboration with DDM

Review Data Access Requirements for Data/Web Services
Design Data/Web Services
Code Data/Web Services
Unit Test Data/Web Services
Implement/Document Data/Web Services
Support System/Integration Testing

PART 12A - Establish Levy Service

Review Business Requirements
Analysis and Design
Code
Unit Test
Implement/Document
Support System/Integration Testing

PART 12B - Establish LEVY Data/Web Services - Collaboration with DDM

Review Data Access Requirements for Data/Web Services
Design Data/Web Services
Code Data/Web Services
Unit Test Data/Web Services
Implement/Document Data/Web Services
Support System/Integration Testing

PART 13A - Establish Installment Agreement Service

Review Business Requirements
Analysis and Design
Code
Unit Test
Implement/Document
Support System/Integration Testing

PART 13B - Establish Installment Agreement Data/Web Services - Collaboration with DDM

Review Data Access Requirements for Data/Web Services
Design Data/Web Services
Code Data/Web Services
Unit Test Data/Web Services
Implement/Document Data/Web Services
Support System/Integration Testing

PART 14A - Update Address Service

Review Business Requirements
Analysis and Design
Code
Unit Test
Implement/Document
Support System/Integration Testing

PART 14B - Establish Address Update Data/Web Services Collaboration with DDM

Review Data Access Requirements for Data/Web Services
Analysis and Design Data/Web Services
Code Data/Web Services
Unit Test Data/Web Services
Implement/Document Data/Web Services
Support System/Integration Testing

6.0 Appendix

6.1 Definitions

BPM – Business Process Management a discipline that encompasses methods, techniques and tools to design, enact, and control business processes involving humans, organizations, applications, documents and other sources of information. BPM leverages tools (software) and solutions (integration) that depict, analyze and optimize business processes and workload management.

BPEL – Business Process Execution Language - an XML-based orchestration language that enables separate businesses to interconnect their applications and share data in distributed or grid computing environment using a combination of Web services.

Business Process – the codification of rules and best practices that constitute the business. It includes people, business services, adapters and some sort of process management activity that manages the flow of work between all the parts.

Business Services - the logical encapsulation of some business function.

Core Services or Infrastructure Services - These are really the plumbing services, which include service management, service communication, policy services, and security services. Security and business policies will often need to incorporate or introduce rules, which is why some of the cores services may be applied together with policy services and security centralization.

Composite Applications – applications built from the business functions of existing applications, with perhaps one or two new components added.

Duplicated SOAs: In Duplicated SOAs things seem to work well. Many services have been duplicated twice or more times. There is little reuse between enterprise development teams. This type of SOA results in high maintenance costs although things may seem to work well.

EDA - Event-Driven Architecture –a software architecture pattern promoting the production, detection, consumption of, and reaction to events. Event-driven architecture complements service oriented architecture (SOA) because services can be started by triggers such as events.

ESB – enterprise service bus - a collection of software components that comprise the foundational services for more complex architectures via an event-driven and standards-based messaging engine (the bus).

Orchestration - a process based approach to combine web services with workflow, typically using BPEL.

SCA – service component architecture, an emerging set of standards related to SOA applications.

Service: A discrete set of business or technical functionality that can be identified, has a defined set of input and output, and is reusable. What goes on behind the service interface is deliberately hidden and should not be of concern to the service consumer.

Shelf-ware SOAs: In Shelf-ware SOAs, SOA is implemented. However, few applications actually use the enterprise services. Enterprise systems are using point-to-point, unstructured integration, and there is little buy-in from several business units within the organization. This SOA is waste of resources and won't deliver benefits. A CoE will help prevent this from happening.

SOA – Service oriented Architecture - a business-centric IT architectural approach to integrate business processes through the implementation of repeatable tasks.

SOA Registry & Repository –a central reference for all the software components within the SOA environment.

SOAP - SOAP messaging is the standard mechanism for communicating with web services, and hides the details of the language of the web service. SOAP messaging is an industry standard, language neutral, vendor neutral, and is platform neutral for *client applications* consuming the web services.

Web Service – web services provide one way of implementing the automated aspects of a given business or technical service and can be used to implement a service-oriented architecture. A major focus of Web services is to make functional building blocks accessible over standard Internet protocols that are independent from platforms and programming languages. These services can be new applications or just wrapped around existing legacy systems to make them network-enabled.

The Wild West SOA: Services proliferate wildly. There is no formal service-definition process. Nobody knows how many services are in place, where they are or what they do. There is no leveraging or reuse of existing resources. This type of SOA environment is extremely difficult to fix and gain control.

WSDL – web service description language - is an XML-based service description that describes how client applications can communicate with the web service endpoints or ports as they are called.

6.2 Industry Best Practices and Trends

- Manage the expectations of SOA investments by understanding that the involved parties don't all envision the same outcome as the objective. Consider these differences in tailoring business communications at all levels.

- SOA is a long-term, complex initiative and organizations must invest in developing the required understanding, best practices, and organizational culture before committing to mission-critical projects. Large projects should be subdivided into smaller components so that the SOA effort is applied initially in a relatively small scope to be expanded over time. Early SOA projects should not last longer than six months from the start of design to the delivery of results. Think strategically, but act tactically. Develop a long-term vision

for SOA, but implement it incrementally, learning during the process and managing the risks of transition.

- Define clear modularization of software layers (e.g., enterprise data and applications, services, business processes, business activity monitoring, etc.).

- As the number of services deployed grows to more than 20 to 30, industry best practices recommend implementing a services repository and ESB.

- The use of standards and meta-data are critical to SOA benefit realization. Use a combination of top-down, bottom-up and business-event driven analysis to identify services — and pay close attention to service granularity.

- Industry best practice is to give preference to platforms that distinctly recognize event and service flows as separate design and deployment models for software, and provide integrated runtime, management and development infrastructure that supports both models.

- The use of design patterns can be used to break down complex problems into manageable chunks that can be developed more efficiently.

- The use of SOA design patterns can map into an Enterprise Service Bus implementations, which can provide components needed for service invocations, routing and transformation of service messages, as well as, facilitating services management.

- These three principles — simplicity, isolation and statelessness — are best practices in the design of all distributed systems, including distributed, interactive SOA due to their inherent complexity.

- Open standards are one of the key principles and benefits of SOA. Standards such as XML, SOAP and WSDL are providing the foundation by which organizations can ensure that a wide variety of enterprise resources can be enabled to interoperate and cooperate as part of an SOA. Standards-based SOA solutions enable organizations to build open, heterogeneous solutions that are not locked into specific vendors or platforms.

- *Business services* are implemented as *web services*. They include the business logic, business rules, and data that make up the business functionality. They can be written in any language that supports web services (Java, .NET, etc.) but they must incorporate a web service interface. SOAP messaging is the standard mechanism for communicating with a web service. This hides the details of the language that the web service is written in. SOAP messaging is an industry standard, language neutral, vendor neutral, and platform neutral.

Some industry standards recommendations, such as WS-I, mandate both SOAP and WSDL in their definition of a Web Service. There are different transport protocols such as HTTP, HTTPS, RPC, SMTP, FTP, and WebSphere MQ that can be implemented. HTTP/HTTPS and SOAP are standards that FTB has the most experience to implement. Part of the FTB IT staff have experience with Distributed Computing Environment (DCE), Distributed Component Object Model (DCOM) and/or RMI (Remote Method Invocation), which are consider distributed object technologies. Object reuse and service orientation are fundamental concepts to these models.

6.2.1 Web Services Data Security, Best Practices, and Trends

Web services technology can be implemented in a variety of ways, which can co-exist with other technologies and can be adopted in an incremental manner without requiring major transformations to legacy applications and/or databases. Many of the features that make web services attractive, including greater accessibility of data, dynamic application-to-application connections are at odds with traditional security models and controls. The good news is that there are solutions, industry standards, and best practices that can be followed. Ensuring the security of web services will involve augmentation of traditional security mechanisms with security frameworks, which are as follows:

- **XML Encryption** - XML Encryption provides confidentiality of web service messages using. XML Encryption is a specification from the World Wide Web Consortium (W3C) and it provides a mechanism to encrypt XML documents.

- **XML Signature** - XML Signatures provide integrity of web service messages. XML Signature is a specification produced to selectively sign XML data in web service messages.

- **Web service authentication and authorization using XACML -** Security Assertion Markup Language (SAML) and XML Access Control Markup Language (XACML) provide mechanisms for authentication and authorization in a Web services environment.

- **Web Services (WS)-Security -** WS-Security is a specification, produced by OASIS. It defines a set of SOAP header extensions for end-to-end SOAP messaging security. It supports message integrity and confidentiality by allowing communicating partners to exchange signed encrypted messages in a Web services environment.

- **Security for Universal Description, Discovery and Integration (UDDI)** - UDDI allows web services to be easily located and subsequently invoked. Security for UDDI enables publishers, inquirers and subscribers to authenticate themselves and authorize the information published in the directory.

6.2.2 Best Practices for a SOA CoE

- The SOA CoE is a partner to project teams and provides a service.
- The SOA CoE must work across all of the SOA domains – business, people, program management, governance, architecture, enabling technologies, operations, and management.
- The SOA CoE is not the sole source of SOA knowledge but manages and communicates it.
- The SOA CoE must be connected to all stakeholders and bridge organizations.
- The SOA CoE must be answerable with defined goals and measures.

6.2.3 XML Gateways and XML Firewalls

A common way to secure web services is to use an XML gateway that receives requests from requesters, performs security checks against the incoming requests, and then forwards the requests to an internal web service provider. XML Gateways are network devices specially designed for XML security with a number of distinct advantages, including performance, security, and reliability. With the continued evolution of SOA, there is a trend in the industry to

move functionality to the network in the form of various hardware devices: XML Gateways or XML Firewalls. As time progresses, more features are being wrapped into these products.

XML Firewalls are essentially high performance proxies, which perform security services such as authentication, authorization, auditing and XML message validation at the message level. They are used to protect backend web services from XML-based threats. A XML security firewall is typically deployed behind an existing IP firewall, and secures all XML traffic before it reaches the web service on the application server. A few years ago, many of the products on the market we labeled “XML Firewalls”; however, the popular industry term being used today is “XML Gateways”, because they are expected to do more than conventional firewalls.

The trend has been repeated in the past, as standards and technology matures additional functionality is added to products. XML Gateways are being adopted by industry and the vendors are taking feedback from the marketplace, customer demands, competitive analysis, and are beefing up XML Gateways with features. Historical trends also bear out the long-term successfulness of the simpler network device approach for network and security functions. IP routing was once done in software. Before Cisco took over with special purpose network devices in the 90’s, the industry debated the relative merits of software, appliance, and the true network device approach for load balancing and SSL acceleration, but now appliance based network devices dominate. It is likely that for XML Security Gateways, the same trend towards purpose-built network appliances will win out; driven by inherently lower cost and greater security required for SOA enabled web services.

6.2.3 Industry Standards for XML based Web Services

OASIS (Organization for the Advancement of Structured Information Standards) is a nonprofit, international consortium whose goal is to promote the adoption of product-independent standards for information formats. The goal of OASIS is not to create structured information standards for XML, but to provide a forum for discussion, to promote the adoption of interoperability standards. The W3C is another nonprofit consortium organization that makes industry recommendations on XML and Web Services standards. A recommendation is a specification that has been approved by OASIS or W3C committee members and made public. This is the highest rating a specification can receive. If a specification is recommended by the OASIS or W3C, chances are it will *become* the standard, if it is not already.

6.2.3.1 XML Web Service Specifications

UDDI 3.0,
XACML 1.0 for Role Based Access Control Policies
SOAP 1.1 with Attachments
WSDL 1.1
XML Signature 1.0
XSLT 1.0
Web Services Security: SOAP Message Security 1.0
Web Services Security: SOAP Message with Attachments (SwA) Profile 1.0
WS-I: Basic Security Profile 1.0
WS-I: Basic Profile 1.1
WS-Security
SAML 2.0
WS-Federation
Liberty Alliance

WS-Trust
XKMS (XML Key Management Specification)
Web Services Notification (WSN) v1.3
WS-Reliability (WS-R) v1.1
WS-ReliableMessaging v1.1
Web Services Resource (WSRF) v1.2
WS-SecureConversation v1.3
Web Services Transaction v1.1

In the early implementation of web services, the specifications above did not exist. Early on there was a mix of proposed recommendations and working draft standards being developed by different vendors that may have slowed down web services adoption. However, with help of OASIS, W3C, and the large vendors such as IBM, Microsoft, BEA, and others web services interoperability and security standards have matured. In the future, it should be expected that the vendors would continue to develop system solutions that will continue to allow system to be interoperable.

6.2.4 Industry Implementation Standards for Web Services Security

Figure 6.2-1 (Current Industry Standards for Implementing Web Services Security) illustrates current industry standards for implementing web services security. The illustration from the Federal government released the *Federal Guide to Web Services Security* (NIST 800-95) shows the building blocks and maps the different standards to the different functional layers found in typical secure web service implementations.

Each of these different implementation standards contains web services data as attributes embedded within well-formed XML protocol structures. For example, WS-Security has a specific XML schema and data attributes embedded within the XML data being passed from the web service provider to the consumer.

Figure 6.2-1: Current Industry Standards for Implementing Web Services Security